PROGRAMME DE CALCUL DES STRUCTURES EN 2D, RDM.ZIP

Auteur : OUAMBO TOBOU Raoul

E-mail : rouambo@yahoo.fr
Mars 2003.
Programme de 6582 octets écrit et testé sur CASIO Graph 30.

SOMMAIRE

CODE DU PROGRAMME

a) Quelques conventions d’écriture

Le signe ! désigne le triangle signifiant ‘Disp’

/ désigne la division

* désigne la multiplication

Les commentaires adressées au lecteur sont précédés de ‘

>= désigne ‘supérieur ou égal’

<= désigne ‘inférieur ou égal’

b) Le code

Programme principal “RDM.ZIP”
Calcul des structures poutres en 2D par éléments finis.

```
ClrText
Locate 1,1,”<<<<<<<RDM.ZIP>>>>>>>”
Locate 8,3,”CALCUL”
Locate 4,4,”DES STRUCTURES”
Locate 4,5,”POUTRES EN 2D”
Locate 1,7,”<<<<<<<<<<<>>>>>>>>>>”
Do
LpWhile Getkey ≠ 31
‘PSEUDO INFINI’        ‘Ce texte entre ‘ et ‘ est un commentaire pour la machine
1e40 → r                         ‘Définition de la valeur du pseudo infini
‘PARAMETRES GRPH’
S-WindAuto
FuncOff
CoordOn
Prog “DONN”
Prog “WAIT”
Prog “ASSM”
Prog “WAIT”
Prog “LIMT”
Prog “WAIT”
Prog “RESO”
Prog “WAIT”
Prog “RSLT”
Prog “WAIT”
Lbl F
ClrText
Locate 1,1,”---FIN DES CALCULS---“       ‘- correspond à la touche “(-)” de la calculatrice
Locate 1,7,”-------RDM.ZIP-------“
Locate 3,3,”DEFORM GLOB (Mat U)”
```

```
Locate 3,4,"REACT GLOB (Mat R)"
Locate 3,5,"VISUALISATIONS"
Locate 3,6,"QUITTER"
3 → X
6 → Y
Prog "GOPT"
A=6 => Prog "ARRT"
ClrText
A=3 => Mat U!
A=4 => MatR!
A=5 => Prog "VISU"
Goto F
```

Sous-programme "DONN"
Saisie des données de la structure à calculer

```
ClrText
Locate 1,7,"-------RDM.ZIP-------"
Locate 1,1,"----MODE DE SAISIE---"
Locate 3,3,"EXTERNE"
Locate 3,4,"INTERNE"
Locate 3,5,"QUITTER"
3 → X
5 → Y
Prog "GOPT"
A=5 => Prog "ARRT"
A=3 => Goto S
ClrText
'ENTRÉE DES NOEUDS'
"NBRE DE NOEUDS" ? → N
Seq(0,X,1,7,1) → List1
List→Mat(List1) → Mat A    'List→Mat est une instruction de l'option List
Mat A → Mat N
For 2→I To N
Augment (Mat N, Mat A) → Mat N
Next
Trn Mat N → Mat N
For 1→I To N
ClrText
"NOEUD"
Locate 7,1,I
"X" ? → Mat N[I,1]
"Y" ? → Mat N[I,2]
ClrText
"FX" ? → Mat N[I,4]
"FY" ? → Mat N[I,5]
"M" ? → Mat N[I,6]
ClrText
```

```
Locate 1,1,"--TYPE DU NOEUD"
Locate 17,1,I
Locate 19,1,"--"
Locate 3,2,"LIA ENC"
Locate 3,3,"APP ENC"
Locate 3,4,"APP ROT"
Locate 3,5,"APP ROU ENC"
Locate 3,6,"APP ROU ROT"
Locate 1,7,"-------RDM.ZIP-------"
2 → X
6 → Y
Prog "GOPT"
A-1 → Mat N[I,3]
If A>=5
Then ClrText
"ANGLE" ? → Mat N[I,7]
IfEnd
Next
'ENTRÉE DES POUTRES'
ClrText
"NBRE DE POUTRES" ? → E
Seq(0,X,1,8,1) → List1
List→Mat(List1) → Mat A
Mat A → Mat E
For 2→I To E
Augment (Mat E, Mat A) → Mat E
Next
Trn Mat E → Mat E
For 1→I To E
ClrText
"POUTRE"
Locate 8,1,I
"NOEUD 1" ? → Mat E[I,1]
"NOEUD 2" ? → Mat E[I,2]
ClrText
"PX" ? → Mat E[I,3]
"PY" ? → Mat E[I,4]
(Mat E[I,3]^2 + Mat E[I,4]^2) ≠ 0 => "GLOB/LOC, 1/0" ? → Mat E[I,5]
ClrText
"PROPRIETES POUTRE"
Locate 19,1,I
"E" ? → Mat E[I,6]
"I" ? → Mat E[I,7]
"S" ? → Mat E[I,8]
Next
Goto C
Lbl S
Dim Mat N → List1
List1[1] → N
```

```
Dim Mat E → List1
List1[1] → E
Lbl C
'INIT DE WAIT'
0 → W
5 → Z
Prog "WAIT"
Text 58,15,"TABLE DES CONNECTIVITES"
'CONNECTIVITES'
Seq(0,X,1,6,1) → List1
List→Mat(List1) → Mat A
Mat A → Mat C
For 2→I To E
Augment (Mat C, Mat A) → Mat C
Next
For 1→I To E
For 1→J To 2
Mat E[I,J] → M
For 1→K To 3
3(M-1)+K → Mat C[3(J-1)+K,I]
Next
Next
Next
Seq(0,X,1,3N,1) → List1
Return
```

Sous-programme "PASS"
Construction de la matrice de passage pour la poutre N° I

```
'CONSTRUCTION DE [T]'
Identity 6 → Mat T
-1 → Mat T[3,3]
-1 → Mat T[6,6]
Mat E[I,1] → A
Mat E[I,2] → B
Mat N[B,1] - Mat N[A,1] → X
Mat N[B,2] - Mat N[A,2] → Y
√(X²+Y²) → L
X/L → X
Y/L → Y
X → Mat T[1,1]
X → Mat T[2,2]
Y → Mat T[1,2]
-Y → Mat T[2,1]
```

```
For 1→X To 2
For 1→Y To 2
Mat T[X,Y] → Mat T[X+3,Y+3]
Next
Next
Return
```

Sous-programme "ELEM"
Calcul des matrices élémentaires de la poutre N° I (rigidité et force)

```
Prog "PASS"
'CONSTRUCTION DE [K]'
Identity 6 → Mat L
Mat E[I,6] → F
Mat E[I,7] → H
Mat E[I,8] → S
FS/L → A
A → Mat L[1,1]
A → Mat L[2,2]
-A → Mat L[1,2]
-A → Mat L[2,1]
FH/(L^3) → A
12A → Mat L[3,3]
-6AL → Mat L[3,4]
-12A → Mat L[3,5]
-6AL → Mat L[3,6]
4AL² → Mat L[4,4]
6AL → Mat L[4,5]
2AL² → Mat L[4,6]
12A → Mat L[5,5]
6AL → Mat L[5,6]
4AL² → Mat L[6,6]
For 3→X To 5
For X+1→Y To 6
Mat L[X,Y] → Mat L[Y,X]
Next
Next
Swap L,2,4
Swap L,2,3
Trn Mat L → Mat L
Swap L,2,4
Swap L,2,3
Trn Mat L → Mat L
'CONSTRUCTION DE {F}'
```

```
[[0][0][0][0][0][0]] → Mat G
If Mat E[I,5] = 1
Then Mat T[1,1] → X
Mat T[1,2] → Y
X Mat E[I,3] + Y Mat E[I,4] → A
-Y Mat E[I,3] + X Mat E[I,4] → B
Else Mat E[I,3] → A
Mat E[I,4] → B
IfEnd
0.5AL → A
A → Mat G[1,1]
A → Mat G[4,1]
0.5BL → B
B → Mat G[2,1]
-BL/6 → Mat G[3,1]
B → Mat G[5,1]
BL/6 → Mat G[6,1]
Return
```

Sous-programme "ASSM"
Assemblage des matrices élémentaires

```
'INITIALISATIONS'
Text 58,15,"                              "     '30 espaces
Text 58,15,"ASSEMBLAGE EN COURS…"
Identity (3N) → Mat K
For 1→I To 3N
0 → Mat K[I,I]
Next
Seq(0,X,1,3N,1) → List2
List→Mat(List2) → Mat F
For 1→I To N
3(I-1) → H
For 1→J To 3
Mat N[I,J+3] → Mat F[H+J,1]
Next
Next
Mat F → Mat R
For 1→I To E
Prog "ELEM"
Trn Mat T * Mat L * Mat T → Mat L
Trn Mat T * Mat G → Mat G
'ASSEMBLAGE'
For 1→X To 6
```

```
Mat C[X,I] → A
For 1→Y To 6
Mat C[Y,I] → B
Mat K[A,B] + Mat L[X,Y] → Mat K[A,B]
Next
Mat F[A,1] + Mat G[X,1] → Mat F[A,1]
Next
Next
'COPIE DE [K]'
Mat K → Mat L
Return
```

Sous-programme "LIMT"
Prise en compte des conditions aux limites aux noeuds

```
Text 58,15,"                              "    '30 espaces
Text 58,15,"CONDITIONS AUX LIMITES"
'CHGMT DE VARIABLES'
For 1→I To N
Mat N[I,3] → X
If X>=4
Then 3(I-1) → H
H+1 → A
H+2 → B
1 → List1[I]
Mat N[I,7] → θ
For 1→Y To 3N
Mat K[Y,A]cosθ + Mat K[Y,B]sinθ → F
-Mat K[Y,A]sinθ + Mat K[Y,B]cosθ → Mat K[Y,B]
F → Mat K[Y,A]
Next
IfEnd
Next
'PENALISATION DIAG'
For 1→I To N
Mat N[I,3] → X
X=1 => Goto 1
3(I-1) → H
H+1 → A
H+2 → B
H+3 → C
If X=2
Then r → Mat K[A,A]
r → Mat K[B,B]
r → Mat K[C,C]
```

```
Goto 1
IfEnd
If X=3
Then r → Mat K[A,A]
r → Mat K[B,B]
Goto 1
IfEnd
r → Mat K[B,B]
X=4 => r → Mat K[C,C]
Lbl 1
Next
Return
```

Sous-programme “RESO”

Résolution du système linéaire [K]{U} = {F}. Fonction d’inversion de matrices de la calculatrice si (nombre de noeuds <= 4), méthode d’élimination de GAUSS sinon)

```
Text 58,15,”                              “      ’30 espaces
Text 58,15,”RESOLUTION EN COURS…”
‘CALCUL DE {U}’
N<=4 => Mat K^-1 * Mat F → Mat U
N>4 => Prog “GAUS”
Identity 1 → Mat K
For 1→I To N
If List1[I] = 1
Then Mat N[I,7] → θ
3(I-1) → H
H+1 → A
H+2 → B
Mat U[A,1]cosθ - Mat U[B,1]sinθ → F
Mat U[A,1]sinθ + Mat U[B,1]cosθ → Mat U[B,1]
F → Mat U[A,1]
IfEnd
Next
‘CALCUL DE {R}’
Mat R + Mat L * Mat U – Mat F → Mat R
Return
```

Sous-programme "RSLT"
Mise en forme des résultats dans Mat U et Mat R

```
Text 58,15,"                              "'30 espaces
Text 58,5,"MISE EN FORME DES RESULTATS"
Mat U → Mat K
Mat R → Mat L
[[0][0][0]] → Mat U
Mat U → Mat G
For 2→I To N
Augment (Mat U, Mat G) → Mat U
Next
Trn Mat U → Mat U
Mat U → Mat R
For 1→I To N
3(I-1) → H
For 1→J To 3
Mat K[H+J,1] → Mat U[I,J]
Mat L[H+J,1] → Mat R[I,J]
Next
Next
Return
```

Sous-programme "GAUS"
Résolution du système linéaire [K]{U} = {F} par la méthode d'élimination de GAUSS avec pivotation partielle. Il y a une première phase de factorisation avant la résolution de deux systèmes triangulaires. Ce programme est donc adapté si l'on désire modifier RDM.ZIP afin de résoudre les cas où les chargements sur la structure sont paramétrés (résolution du système pour plusieurs valeurs de {F}).

```
'FACTORISATION'
3N → M
Seq(X,X,1,M,1) → List2
List→Mat(List2) → Mat U
For 1→K To M-1
List2[K] → X
Abs (Mat K[X,K]) → X
K → Y
For K+1→I To M
List2[I] → L
If (Abs Mat K[L,K]) > X
Then Abs Mat K[L,K] → X
I → Y
IfEnd
Next
```

```
List2[K] → I
List2[Y] → List2[K]
I → List2[Y]
List2[K] → L
1/Mat K[L,K] → P
L → H
For K+1→ I To M
List2[I] → L
P Mat K[L,K] → C
C → Mat K[L,K]
For K+1→J To M
Mat K[L,J] – C Mat K[H,J] → Mat K[L,J]
Next
Next
Next
'RESOLUTION'
List2[1] → L
Mat F[L,1] → Mat U[1,1]
For 2→I To M
0 → C
List2[I] → L
For 1→J To I-1
C + Mat K[L,J] * Mat U[J,1] → C
Next
Mat F[L,1] – C → Mat U[I,1]
Next
List2[M] → L
Mat U[M,1] / Mat K[L,M] → Mat U[M,1]
For M-1→I To 1 Step –1
0 → C
List2[I] → L
For I+1→J To M
C + Mat K[L,J] * Mat U[J,1] → C
Next
(Mat U[I,1] – C) / Mat K[L,I] → Mat U[I,1]
Next
Return
```

Sous-programme "VISU"
Visualisations graphique des résultats

```
'INITIALISATIONS'
Dim Mat N → List2
List2[1] → N
Dim Mat E → List2
List2[1] → E
ClrText
"N° DE POUTRE" ? → I
Cls
Text 25,5,"PREPARATION DES DONNEES…"
[[0][0][0][0][0][0]] → Mat F
50 → r
S-Gph2 DrawOff
S-Gph3 DrawOff
S-Gph1 DrawOn,xyLine,List1,List2,1,Dot
For 1→J To 2
3(J-1) → H
Mat E[I,J] → X
For 1→K To 3
Mat U[X,K] → Mat F[H+K,1]
Next
Next
Prog "PASS"
Mat T * Mat F → Mat F
If Mat E[I,5] = 1
Then Mat T[1,1] → X
Mat T[1,2] → Y
X Mat E[I,3] + Y Mat E[I,4] → A
-Y Mat E[I,3] + X Mat E[I,4] → B
Else Mat E[I,3] → A
Mat E[I,4] → B
IfEnd
 B / (24Mat E[I,6] * MatE[I,7]) → K
-Mat F[3,1] → D
Mat F[2,1] → F
(3Mat F[5,1] + L Mat F[6,1] + KL^4 – 2DL – 3F) / L² → C
(-Mat F[6,1] – D – 2CL – 4KL^3) / (3L²) → W
Mat F[1,1] → G
-A / (2Mat E[I,6] * Mat E[I,8]) → H
(Mat F[4,1] – G – HL²) / L → T
Lbl D
ClrText
Locate 1,1,"----VISUALISATIONS---"
Locate 3,3,"DEPLACEMENTS"
Locate 3,4,"EFFORTS"
Locate 3,5,"PTS/GRPH"
```

```
Locate 13,5,r
Locate 3,6,"QUITTER"
Locate 1,7,"-------RDM.ZIP-------"
3 → X
6 → Y
Prog "GOPT"
A=6 => Return
If A=5
Then ClrText
"NBRE DE PTS" ? → r
Abs r → r
r > 255 => 255 → r
Goto D
IfEnd
L / (r-1) → P
Seq(X,X,0,L,P) → List3
If A=3
Then Prog "DEPL"
Goto D
IfEnd
Prog "EFFT"
Goto D
```

Sous-programme "DEPL"

Visualisation des déplacements dans le repère local de la poutre N° I.

```
Cls
Text 30,30,"UN INSTANT SVP"
Seq(HX² + TX + G,X,0,L,P) → List4
Seq(KX^4 + WX^3 + CX² + DX + F,X,0,L,P) → List5
Seq(4KX^3 + 3WX² + 2CX + D,X,0,L,P) → List6
Lbl A
ClrText
Locate 1,1,"-----REPERE LOCAL----"
Locate 3,3,"DEFORMEE"
Locate 3,4,"DEPL X"
Locate 3,5,"DEPL Y"
Locate 3,6,"ROTATIONS"
Locate 3,7,"QUITTER"
3 → X
7 → Y
Prog "GOPT"
A=7 => Return
If A=3
Then List3 + List4 → List1
```

```
List5 → List2
Else List3 → List1
A=4 => List4 → List2
A=5 => List5 → List2
A=6 => List6 → List2
IfEnd
Prog "TRCE"
Goto A
```

Sous-programme "EFFT"

Diagrammes des efforts internes de la poutre N° I.

```
Cls
Text 30,30,"UN INSTANT SVP"
List3 → List1
Mat E[I,6] * Mat E[I,8] → Z
Seq(Z(2HX + T),X,0,L,P) → List4
Mat E[I,6] * Mat E[I,7] → Z
Seq(-Z(24KX + 6W),X,0,L,P) → List5
Seq(Z(12KX^2 + 6WX + 2C),X,0,L,P) → List6
Lbl B
ClrText
Locate 1,1,"---EFFORTS INTERNES--"
Locate 3,3,"NORMAL"
Locate 3,4,"TRANCHANT"
Locate 3,5,"MOMENT FLECHISSANT"
Locate 3,6,"QUITTER"
Locate 1,7,"-------RDM.ZIP-------"
3 → X
6 → Y
Prog "GOPT"
A=6 => Return
A=3 => List4 → List2
A=4 => List5 → List2
A=5 => List6 → List2
Prog "TRCE"
Goto B
```

Sous-programme “TRCE”

Tracé et mise en forme des graphiques

```
Int(r/12) → θ
Frac(r/12) >= 0.5 => θ+1 → θ
0 → X
0 → Y
List2[1] ≠ 0 => 1 → X
For 2→Z To r
List2[Z] ≠ 0 => 1 → X
List2[Z] ≠ List2[Z-1] => 1 → Y
Next
If X=0
Then ClrText
Locate 3,5,”NUL PARTOUT”!
Return
IfEnd
Y=0 => 0 → List2[1]
DrawStat
For 1→Z To r Step θ
r – Z < θ => r → Z
0 → Y
List2[Z] > 0 => Ymin > 0 => Ymin → Y
List2[Z] < 0 => Ymax < 0 => Ymax → Y
F-Line List1[Z],Y,List1[Z],List2[Z]
Next
Plot!
Return
```

Sous-programme “ARRT”

Affichage de la fenêtre de sortie

```
ClrText
Locate 1,1,”<<<<<<<RDM.ZIP>>>>>>>”
Locate 5,3,”VERSION 2.1”
Locate 3,5,”(C) OUAMBO 2003”
Locate 1,7,”<<<<<<<<<<<>>>>>>>>>>”
Stop
```

Les programmes “GOPT” et “WAIT” qui suivent ont été conçues pour améliorer la convivialité de n’importe quel autre programme. C’est à cette fin qu’elles ont été utilisées dans RDM.ZIP.

Sous-programme "GOPT"
Permet un choix d'option par pointeur, dans un menu affiché à l'écran.

```
X → A
Locate 2,A,"→"
Do
A → B
Getkey = 28 => A-1 → A
Getkey = 37 => A+1 → A
A = X-1 => X → A
A = Y+1 => Y → A
A ≠ B => Locate 2,B," "        'un espace
A ≠ B => Locate 2,A,"→"
LpWhile Getkey ≠ 31
Return
```

Pour utiliser "GOPT" dans un programme:

- Avec des instructions d'affichage (comme Locate), afficher une page de menu à l'écran. Les options à choisir doivent être placées les unes à la suite des autres, et commencer au moins à la 3ème colonne (du mode caractère) de l'écran.
- Mettre le N° de la première ligne où se trouve les options dans la variable X, et celui de la dernière dans la variable Y.
- Appeler le programme "GOPT", qui s'exécute en affichant le menu et un pointeur de choix à l'utilisateur. Ce dernier utilisera alors les touches de direction pour choisir une option et la valider avec la touche EXE.
- Le N° de la ligne où se trouve l'option choisie est renvoyée dans le programme principal par la variable A.

Le code de RDM.ZIP présente plusieurs exemples de création de menus par "GOPT".

Sous-programme "WAIT"
Crée un objet graphique permettant de suivre l'avancement d'un programme.

```
W ≠ 0 => Goto 1
Cls
AxesOff
ViewWindow 0,6Z/5,Xscl,0,50,Yscl
Z/10 → W
F-line W,20,11W,20
F-line W,30,11W,30
F-line W,20,W,30
F-line 11W,20,11W,30
Text 17,27,"UN INSTANT SVP !"
Goto2
Lbl 1
For W→O To W+1 Step Z/110        'Attention, ne pas confonfre O (Orange) et 0 (1 –1=0) ici
```

```
F-line O,20,O,30
Next
W+1 → W
Lbl 2
Return
```

Pour utiliser “WAIT” dans un programme:

- Initialiser “WAIT” : mettre 0 dans W (0 → W), mettre le nombre d’étapes à suivre jusqu’à la fin du programme dans Z, et appeler le programme “WAIT”.
- Pendant l’exécution d’une étape, on peut si l’on veut (comme dans RDM.ZIP) afficher à l’écran des messages d’évolution à l’aide de l’instruction Text x,y,”????”, placé à l’endroit du code où commence l’étape en question. A l’endroit où l’étape se termine, placer l’instruction Prog “WAIT”. L’effaçage d’un texte affiché se fait à l’aide de l’instruction Text x,y,”   “, où l’on mettra autant d’espaces que nécessaire pour effacer l’ancien message.

Cette démarche a été effectuée dans le code de RDM.ZIP. Quand on utilise “WAIT” dans un programme principal, il ne faut pas modifier les variables W et Z tant que les étapes à suivre ne sont pas terminées.

MANUEL D'UTILISATION DE RDM.ZIP

I- A PROPOS DE RDM.ZIP

RDM.ZIP est un programme de calcul des structures poutres en deux dimensions. Cette version 2.1 a été conçue pour offrir un meilleur compromis entre taille du programme et fonctionnalités. Le programme présente une modularité accrue, ce qui rend sa personnalisation aisée.

II- TECHNIQUES DE CALCUL UTILISEES

RDM.ZIP calcule les structures à l'aide de la méthode des éléments finis. Un seul type d'élément est utilisé dans cette version. C'est l'élément résultant de l'association :

- D'un élément barre formulé en déplacement, pour la prise en compte des effets d'extension.
- D'un élément poutre formulé en déplacement, sans prise en compte du cisaillement transversal (modèle de Navier-Bernouilli), permettant de représenter les effets de flexion.

L'élément ainsi crée est appelé « POUTRE » dans RDM.ZIP, et est caractérisé par :

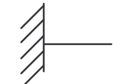
- Son module de Young E
- Le moment quadratique de sa section I
- L'aire de sa section S

. Une poutre est terminée par des nœuds à ses deux extrémités, et peut avoir ou non une charge répartie à densité constante. On ne peut définir une charge localisée sur une poutre, mais plutôt sur un nœud. Ainsi, lors de la modélisation d'une structure en vue de son calcul, un nœud pourra être :

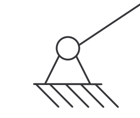
- Une liaison : c'est le cas où il sert de jonction à deux poutres. Il existe plusieurs types de liaisons (encastrement, rotule, avec excentricité, …), mais cette version de RDM.ZIP ne prend en compte que les liaisons encastrement (nommée LIA ENC dans le programme). Ce type de liaison réalise la continuité des déplacements et des rotations à la jonction de deux poutres, et peut par exemple être une soudure des poutres ou le point d'application d'une charge localisée sur un profilé, qui a conduit à le considérer comme deux poutres dans la modélisation. La prise en compte d'autres types de liaison augmenterait nettement la complexité (introduction des techniques de condensation) et la taille du programme. C'est pourquoi elles n'ont pas été introduites dans cette version qui se veut aisément portable et pédagogique.

- Un appui : c'est une jonction de la structure avec le milieu extérieur. RDM.ZIP prend en compte 4 types d'appui :

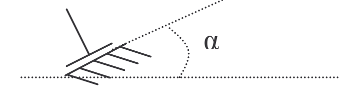
  - Appui encastré (APP ENC)



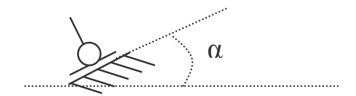
  - Appui rotule (APP ROT)



  - Appui roulant encastré ou glissière (APP ROU ENC)



  - Appui roulant rotule ou glissière avec rotule (APP ROU ROT)



L'assemblage dans RDM.ZIP est réalisé par rigidité directe. Les conditions aux limites sont prises en compte par pénalisation diagonale, où la valeur du pseudo-infini est 1e40 (modifiable dans le code du programme).

La résolution du système final est faite par la fonction d'inversion de matrices de la calculatrice si la taille du système est inférieure ou égale à 12 (4 nœuds dans la structure). Dans le cas contraire, le système est résolu par la méthode d'élimination de Gauss.

Le listing du programme précise la fonction de chaque sous-programme. La simplicité du style de programmation permet une compréhension aisée, et facilite ainsi la personnalisation de RDM.ZIP.

III- UTILISATION DE RDM.ZIP

a) Paramétrage de la calculatrice

L'utilisation de RDM.ZIP n'exige aucun paramétrage particulier de la calculatrice. Par contre, le paramétrage suivant est réalisé en cours d'exécution :

- S-WindAuto
- FuncOff
- CoordOn
- S-Gph2 DrawOff
- S-Gph3 DrawOff
- S-Gph1 DrawOn,xyLine,List1,List2,1,Dot

b) Saisie des données

Les données d'entrée du programme résident dans les matrices Mat N (pour les nœuds) et Mat E (pour les poutres). Deux modes de saisie des données dans ces matrices sont alors possibles :

- Saisie externe : ici, les matrices Mat N et Mat E sont remplies dans le menu des matrices comme suit :

  - Mat N (les nœuds) : sa dimension est Nx7 (N = nombre de nœuds), et son contenu celui du tableau çi-dessous.

X	Y	Type	Fx	Fy	M	α

La ligne N° I contient les informations sur le nœud N° I.
X et **Y** sont les coordonnées du nœud dans le repère global.
Type est son type, et peut prendre les valeurs 1,2,3,4,5 telles que :

- 1 => Liaison encastrement
- 2 => Appui encastré
- 3 => Appui rotule
- 4 => Appui roulant encastré
- 5 => Appui roulant rotule

{**Fx**, **Fy**, **M**} représente l'action localisée exercée sur le nœud, exprimé dans le repère global de la structure.
α représente l'angle de la ligne de glissement avec l'axe des x du repère global, dans le cas des appuis roulants.

- Mat E (les poutres) : sa dimension est Ex8 (E = nombre de poutres), et son contenu est :

Nœud 1	Nœud 2	Px	Py	Glob/Loc, 1/0	E	I	S

La ligne N° I contient les informations sur la poutre N° I.
Nœud 1 est le N° du nœud de la première extrémité de la poutre (origine du repère local de la poutre), **Nœud 2** celui de la seconde extrémité.
Px et **Py** représentent les densités suivant x et y de l'effort réparti sur la poutre.
Glob/Loc prend la valeur 1 si Px et Py sont donnés dans le repère global, et 0 s'ils le sont dans le repère local.
E, I, S sont les propriétés de la poutre : module de Young (E), moment quadratique de la section (I), aire de la section (S).

- Saisie interne : dans ce mode, toutes les données ci-dessus sont tour à tour demandées par la calculatrice, jusqu'au remplissage des matrices Mat N et Mat E.

N.B. : quel que soit le mode de saisie, les matrices Mat N et Mat E ne sont pas modifiées après l'exécution du programme. La reprise des calculs après modification de quelques paramètres est donc facilitée.

c) Les unités utilisées

Les unités dans les calculs sont celles avec lesquelles les données ont été saisies, et l'unité d'angle est bien sûr celle paramétrée dans la calculatrice. Ainsi, si vous utilisez un système d'unités où les longueurs sont en mm et les forces en daN, alors les résultats seront dans le même système, à l'exception de l'unité d'angle qui sera toujours le radian, quelle que soit l'unité d'angle utilisée lors de la saisie des données.

d) Déroulement des calculs

Une interface graphique conviviale permet de suivre l'évolution des calculs. Cette fonctionnalité, ajoutée à la grande modularité du code offre a RDM.ZIP une envergure pédagogique.

e) Affichage des résultats

Après les calculs, un menu des résultats offre les options suivantes :

- Déplacements globaux : il donne dans le repère global, par nœud, les déplacements suivant x (u), suivant y (v) et la rotation (θ). Ces informations sont fournies par la matrice Mat U, consultable même après arrêt du programme.

u	V	θ

La ligne N° I contient les données du nœud N° I. A cause de la valeur du pseudo-infini de pénalisation prise à 10E40, un effort de l'ordre de 10E-40 sera considéré comme nul.

- Réactions globales : cette option est similaire à la précédente. Elle affiche la matrice Mat R contenant les efforts globaux de chaque nœud ; Fx, Fy, Mz.

Fx	Fy	M

A cause de la valeur du pseudo-infini de pénalisation prise à 10E40, un effort de l'ordre de 10E-10 sera considéré comme nul.

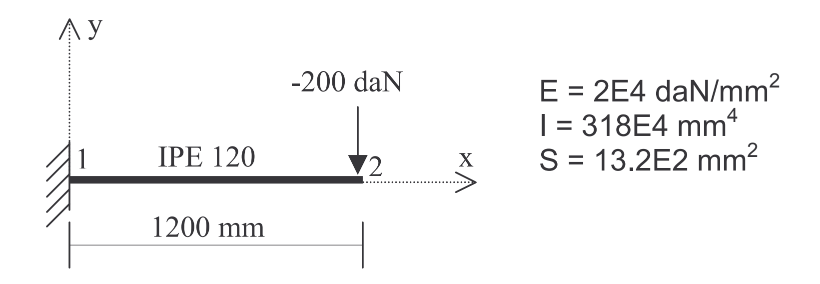
- Visualisations : cette option ouvre un menu de visualisation graphique des résultats. Toutes les visualisations sont relatives à une poutre donnée, et exprimées dans son repère local. Les options proposées sont :
  - Déplacements : pour avoir la déformée, les déplacements suivant x, suivant y et les rotations sur la poutre.
  - Efforts : pour avoir les diagrammes des efforts internes sur la poutre (effort normal, tranchant et moment fléchissant).
  - Nombre de points par graphe : ce nombre est par défaut de 50, et est modifiable par cette option.

  Si les matrices Mat N, Mat E, Mat U, Mat R et sont pas modifiées après l'arrêt du programme, on peut revoir les visualisations à tout moment, en exécutant le sous-programme VISU indépendamment. Pendant un affichage graphique, la fonction « Trace » de la calculatrice

permet de se déplacer sur les points du graphe. La touche « EXE » permet de quitter la visualisation en cours et revenir au menu précédent.

IV- EXEMPLES DE PROBLEMES RESOLUS AVEC RDM.ZIP

a) Calcul d'une poutre encastrée



Les données relatives à cette structure sont :

- Les nœuds (Mat N)

X	Y	Type	Fx	Fy	M	α
0	0	2	0	0	0	0
1200	0	1	0	-200	0	0

- Les poutres (Mat E)

Nœud 1	Nœud 2	Px	Py	Glob/Loc, 1/0	E	I	S
1	2	0	0	0	2E4	318E4	13.2E2

Dans un premier temps, lancez le programme et choissez le mode de saisie interne. Le nombre de nœuds est 2 et le nombre de poutres 1. Le type du nœud n°1 est APP ENC, et celui du nœud n°2 est LIA ENC.

Après calcul, vérifiez alors les résultats suivants :

- Les déplacements globaux

u	v	θ
0	-2.00E-38	-2.40E-35
0	-1.81	-2.26E-3

u et v sont en mm, θ en radian.

- Les réactions globales

Fx	Fy	M
0	200	240000
0	-200	0

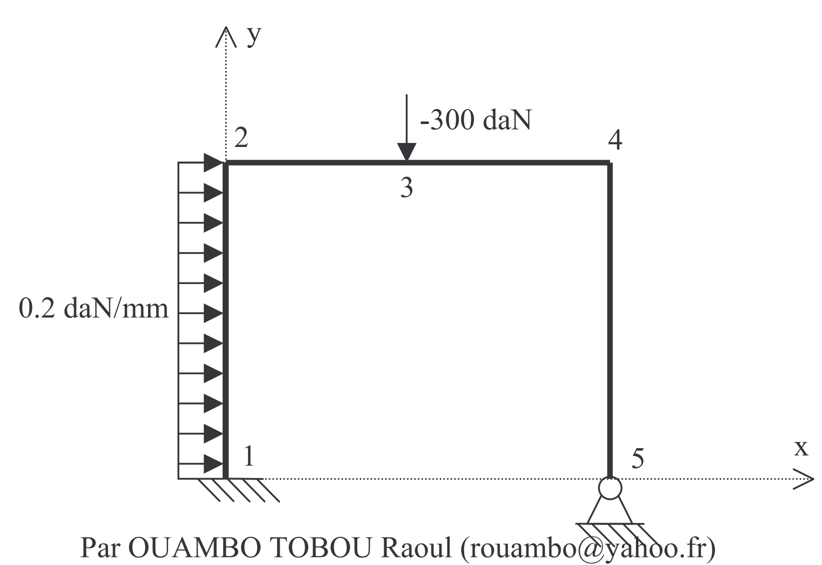
Fx et Fy sont en daN, M en mm.daN

- Les visualisations

Utilisez ce menu des visualisations pour avoir l'allure des grandeurs internes de la poutre. Avec la fonction « Trace » (F1) de la calculatrice, appréciez alors l'évolution de ces grandeurs sur la poutre. Elles doivent être en accord avec les résultats ci-dessus. Appuyez sur EXE pour quitter une visualisation et retourner au menu précédent.

Vous pouvez maintenant sortir du programme aller dans le menu des matrices changer par exemple l'amplitude de la force Fy à +200 daN (force orientée vers le haut) dans Mat N. Exécuter alors de nouveau RDM.ZIP en choisissant un mode de saisie externe. Comparer alors les résultats des deux chargements.

b) Calcul d'un portique



Les poutres ici sont identiques en section et longueur à la poutre précédente.

Les données relatives à cette structure sont :

- Les nœuds (Mat N) :

X	Y	Type	Fx	Fy	M	α
0	0	2	0	0	0	0
0	1200	1	0	0	0	0
600	1200	1	0	-300	0	0
1200	1200	1	0	0	0	0
1200	0	3	0	0	0	0

- Les poutres (Mat E) :

Nœud 1	Nœud 2	Px	Py	Glob/Loc, 1/0	E	I	S
1	2	0.2	0	1	2E4	318E4	13.2E2
2	3	0	0	0	2E4	318E4	13.2E2
3	4	0	0	0	2E4	318E4	13.2E2
4	5	0	0	0	2E4	318E4	13.2E2

Entrer dans le menu des matrices, et saisissez dans les matrices Mat N (5x7) et Mat E (4x8) les contenus ci-dessus.

Lancez alors RDM.ZIP, et choisissez le mode de saisie externe. Vérifiez alors que vous obtenez les résultats suivants.

- Les déplacements globaux (Mat U)

u	v	θ
1.93E-38	-9.88E-39	-8.26E-36
0.333	-4.49E-3	-2.81E-4
0.332	-0.103	-4.44E-5
0.331	-9.14E-3	7.99E-5
4.72E-39	-2.01E-38	-4.54E-4

- Les réactions globales (Mat R)

Fx	Fy	M
-192.84	98.82	82584.95
-8E-10	-7.9E-12	-1.2E-9
2E-10	-300	1E-10
1.43E-10	-7.7E-12	2.3E-9
-47.16	201.18	-2E-10

Les unités sont les mêmes que celles du précédent problème. Vous pouvez aussi visualiser l'évolution des grandeurs internes dans chacune des poutres.

Comme il a été dit plus haut, n'oubliez pas que dans les deux exemples traités, la valeur d'une composante de déplacement rigoureusement nulle est de l'ordre de 1E-40, et celle d'une composante de réaction est de l'ordre de 1E-10.